

DOCUMENT RESUME

ED 410 936

IR 018 498

AUTHOR Thibeault, Nancy E.
TITLE The Web-Database Connection Tools for Sharing Information on the Campus Intranet.
PUB DATE 1997-00-00
NOTE 11p.; In: Association of Small Computer Users in Education (ASCUE) Summer Conference Proceedings (30th, North Myrtle Beach, SC, June 7-12, 1997); see IR 018 473.
PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Software Evaluation; *Computer System Design; *Databases; Design Requirements; Higher Education; Information Technology; *World Wide Web
IDENTIFIERS Microsoft Access; Microsoft Windows; *Web Pages; Web Sites

ABSTRACT

This paper evaluates four tools for creating World Wide Web pages that interface with Microsoft Access databases: DB Gateway, Internet Database Assistant (IDBA), Microsoft Internet Database Connector (IDC), and Cold Fusion. The system requirements and features of each tool are discussed. A sample application, "The Virtual Help Desk" demonstrates how to interact with a database using the four tools. Conclusions are then drawn about the four tools. DB Gateway is too limited and too complex to be a useful tool. For a user who wants a free product and does not have Windows NT server, IDBA appears to be an acceptable alternative. IDC is recommended for users who own a Windows NT server. Even with the disadvantage of the file organization, IDC is worth further investigation. It is still fairly straightforward and has the added benefit of the Access IIS Add-In to automatically generate .idc and .htx files. Cold Fusion is worth the investment if a Web programming development language is needed. Features of the four tools are summarized in a table at the end of the paper. (AEF)

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

The Web-Database Connection

Tools for Sharing Information on the Campus Intranet

Nancy E. Thibeault
Computer Services Manager
Miami University
4200 East University Blvd.
Middletown, OH 45042
phone: (513) 727-3355
fax: (513) 727-3223
thibeane@muohio.edu

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

- ☐ This document has been reproduced as received from the person or organization originating it.
- ☐ Minor changes have been made to improve reproduction quality.

* Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY

C.P. Singer

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)

Introduction

The World Wide Web (WWW, the Web) is growing rapidly. In a few short years, it has grown from a means of displaying static documents to a medium for sharing dynamic information. Unfortunately, most Web sites are not taking full advantage of the Web's dynamic information sharing capability.

Not long ago, adding dynamic content to your Web pages meant learning a CGI (Common Gateway Interface) programming language and writing custom scripts, a tedious and time-consuming process. All this has changed, new Web database tools have been developed that provide easy access to databases from Web pages. With these tools, you can create dynamic, data-driven Web sites, instead of static pages with fixed text and images. These tools allow the user to interact with the database through HTML forms. The user can retrieve, store, update, and delete data from the database using the Web interface. Simply update your database and the new information is immediately available on the Web. Using a Web browser interface to your databases provides your users with an easy-to-use, platform-independent interface.

This paper evaluates four tools for creating Web pages that interface with Microsoft Access databases: DB Gateway, Internet Database Assistant, Microsoft Internet Database Connector, and Cold Fusion. The system requirements and features of each tool are discussed. A sample application, "The Virtual Help Desk" demonstrates how to interact with a database using the four tools. Some of the tools include Wizards to assist you in your development efforts; however, you will need to become familiar with Hypertext Markup Language (HTML), including Forms and Tables, relational databases, and Structured Query Language (SQL). Some programming experience is also helpful.

Background

Miami University Middletown is a branch campus with approximately 2300 students, 75 full-time faculty members, 75 part-time faculty members and 150 full-time staff members. My department is responsible for all computer services on campus. I have 2.5 staff members and 15 student workers. We maintain a campus network consisting of 350 microcomputers (Windows '95, Windows 3.1, and Macintosh), five servers (Novell, Windows NT, LINUX, and two VMS), email

and an Internet connection. We are also responsible for supervising the computer center and five computer classrooms, providing faculty and student training, plus completing approximately 200 work orders per month.

For the past two years, I have used an Access database to track the work orders. All calls that could not be handled over the phone were recorded in the database and assigned to a worker. Work lists of open calls were printed daily for each worker. When jobs were completed the workers marked them on their lists then I would update the database. This system required a lot of my time and had several problems. Work requests would be phoned in, but would not be passed on, or would get lost before they were entered into the database. I normally updated work lists at the end of the day. It was difficult to keep the work lists current. After updating and printing the work lists, I often received additional work requests. I would add the new requests to the database and reprint the work lists. After the workers picked up their work lists, additional work requests would be received and the workers would not see the requests until I reprinted their lists, delaying the processing of the request.

In addition to my Computer Services duties, I am also required to teach one course per semester. In the fall of '96, instead of teaching one course for the entire semester, I covered three classes for the last five weeks of the semester while a faculty member went on maternity leave. My staff members were concerned that they would have the additional burden of maintaining the work lists while I was busy teaching, and asked me to try to automate the system. Using a Web interface seemed like a natural solution, so I began my search for products to create a Web interface to the existing Access database. The result was the Virtual Help Desk. Users can add and update work orders, print work lists, and check the status of work orders via the Web interface.

The Virtual Help Desk was initially written using DB Gateway then rewritten in Cold Fusion. It is currently only used by the Computer Center staff and student workers, but will be made available campus wide in the fall. This system has saved me a lot of time and headaches. The workers really like using the system. They can enter new work orders and check their work lists from anywhere. They can even dial up from home. I still handle the task of updating the status of the work orders, the workers would like to do this themselves; however, my definition of "Job Completed" is often different from theirs.

The Database Format

The Virtual Help Desk database consists of two main tables, Directory and NewWork. A unique ID field (EmpID) links the two tables. The Directory table includes an entry for every faculty member, staff member and computer classroom. The NewWork table contains the associated work orders. A third table, Workers, contains the names of the current Computer Center staff and student workers.

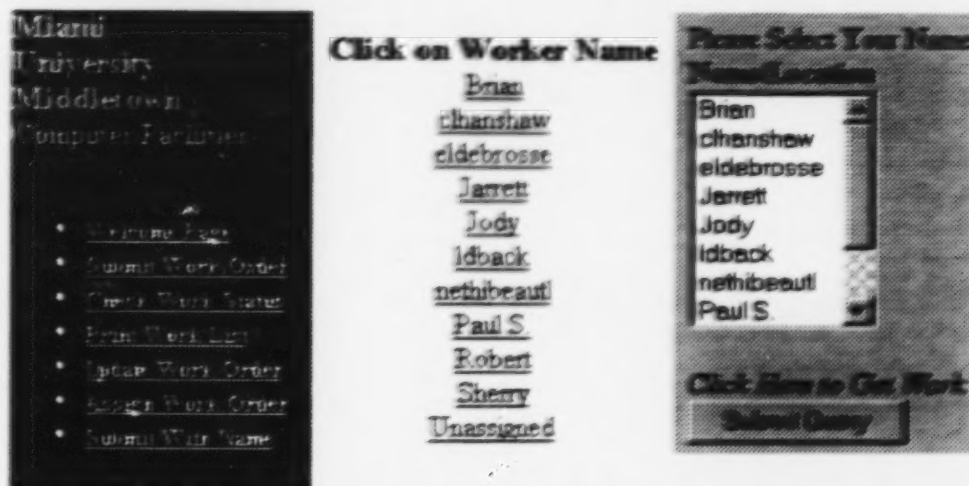
Directory	New Work	Workers
EmpID	EmpID	WorkerID
LastName	Date	WorkerNm
FirstName	Problem	
Department	DueDate	
Bldg	AssignedTo	
Room	Status	
Phone	CompletionDate	
	RefNum	

The Web Interface

The current interface uses frames. The smaller frame contains the menu and is always visible. The larger frame is used to show the results of selecting a menu option.

Figure 1 shows the menu and the results of selecting the "Print Work List" option. Two different methods are shown. I initially used a selection list and submit button, but later changed to using the name as a hyperlink to make it easier for the user.

Figure 1. Virtual Help Desk Menu showing two different methods for "Print Work List"



When "Print Work List" is selected from the menu, the list of worker names appears in the larger frame. The user clicks on a name (and the submit button if using selection list), then the work list appears in the larger frame. Figure 2 shows a sample work list.

Figure 2. Sample Work List

Brian's Work List For Sun March 23, 1997

Last Name	Office	Phone	Date	Ref Num
Schubert	114 JH	217	11/6/96	2250
She selects Start Find File then gets illegal operation error and computer shuts down.				
Capo	205 TH	266	11/6/96	2247
Computer still locking up intermittently - called back into Dell 11/6/96 - wants us to call back from machine - Brian removed cache - goes fast and slow when in screen saver (even after removing the cache). Mariene will let us know if it locks up. If we haven't heard from her by Wed call back to Dell. When jerks, colors fade.				
Johnson	114 JH	117	11/11/96	2272
Reinstall Reflections on all the admissions machines except Dave's and Mary Lu's (it works on theirs, but not on the others)				
Lynch	213 JH	259	11/13/96	2289
Install multimedia kit in her computer - kit in Nancy's office				

The Web Database Tools

SQL is used to communicate with relational databases. HTML does not support embedded SQL; however, the Web database tools add this capability. The embedded SQL is interpreted by the tool, which formulates a query, contacts the database, collates the query results and presents the results to the user in an HTML document.

The system requirements and functions are shown in Figure 3. In the following sections, I show a sample of the files and the code needed by each tool to select an employee and print a work list.

Figure 3. System Requirements for Web Database Tools

Tool	Operating System	Web Server	Supported Databases	Method	Functions
DB Gateway	Windows '95 Windows NT	O'Reilly's WebSite Netscape Enterprise Server	Microsoft Access Microsoft FoxPro	Embed SQL queries in hidden Form elements	Queries Add record
Intranet 2001 IDBA		Not needed	ODBC	Embed SQL queries in comments	Any SQL query
Microsoft IDC	Windows NT	IIS	ODBC	IDC and HTX files	Any SQL query
Cold Fusion	Windows '95 Windows NT	IIS	ODBC	Cold Fusion Mark Up Language	SQL queries plus its own programming language

DB Gateway

Initially, I searched for freeware. The first product I experimented with was DB Gateway. DB Gateway is a CGI application that provides World Wide Web access to Microsoft Access and FoxPro databases. It was developed by Computer Systems Development Corporation. It is available for download at <http://fcim1.csdc.com/DBGate/dbintro.htm>.

I used DB Gateway to implement my first version of the Virtual Help Desk. This version was created in two weekends. We used this version for three months. The Web interface was only used to add new work orders and print work lists. The Access database interface continued to be used for work order assignment and update. This system was fairly primitive, but it achieved the goal of allowing the workers to enter work orders and print work lists.

DB Gateway uses hidden HTML form elements to build the SQL queries and report templates to display the results of the query. The WKLIST code below includes the hidden HTML form elements that will be used to generate the SQL query to select the WorkerIDs and WorkerNms from the Workers table. When the user clicks on the "Print Work" option, the SQL query is sent to DB Gateway, where it is processed. The result of the query is displayed using the indicated report template, PRINTWK.

```
<FORM METHOD="POST" ACTION="/cgi-win/dbgate.exe">
<INPUT type="hidden" name="ACTION" value="SubmitExternalQuery">
<INPUT type="hidden" name="DBNAME" value="MUMCCDB">
<INPUT type="hidden" name="DBTYPE" value="ACCESS">
<INPUT type="hidden" name="QUERYNAME" value="WorkList">
<INPUT type="hidden" name="DBACTION" value="Query">
<INPUT type="hidden" name="FIELDS0" value="WorkerID, WorkerNm">
<INPUT type="hidden" name="FROM0" value="Workers">
<INPUT type="hidden" name="ORDERBY0" value="WorkerNm">
<INPUT type="hidden" name="REPORTNAME" value="PRINTWK">
<INPUT type="image" name="" Src="/print.gif">
</FORM>
```

The PRINTWK report template below displays a selection list of worker names that were generated from the WKLIST code above. PRINTWK also formulates the query to print the work list for the selected worker. The SQL query in PRINTWK uses an enumerated name designation that splits the WHERE clause into three hidden form variables, WHERE0, WHERE1, and WHERE2. DB Gateway will concatenate the various parts into sequential order to form the SQL statement.

```
<INPUT type="hidden" name="FIELDS0" value="RefNum, Date, LastName, Problem">
<INPUT type="hidden" name="FROM0" value="Directory, NewWork, Workers">
<INPUT type="hidden" name="WHERE0" value="WorkerNm = AssignedTo and Status =
'Waiting' and Directory EmpID = NewWork EmpID and ">
```

```
<INPUT type="hidden" name="WHERE1" value=WorkerID = ">
<SELECT NAME="WHERE2" >
<OPTION VALUE=" <FIELD NAME="WorkerID"> <FIELD NAME="WorkerNm">
</SELECT>
```

When the user selects a name from the list and clicks the submit button, DB Gateway formulates the SQL query, sends the query to the Access database, then displays the result using a default format. I could have defined another report template, but elected to use the default.

Internet Database Assistant

Internet Database Assistant (IDBA) is a product of Intranet 2001, Inc. It provides SQL access to any ODBC data sources from your Web page. The freeware version is available for download at <http://www.me2001.com>. There is also a commercial version, which was not investigated. IDBA does not require a Web server.

IDBA is an improvement over DB Gateway. IDBA queries appear as embedded comments in the same HTML form where they are used. IDBA is much simpler than DB Gateway, I rewrote the Virtual Help Desk using IDBA in an afternoon. We never used this implementation, because I went on to discover a product that I liked better.

The code below queries the database and displays the selection list of worker names. Comments are used to send the queries to IDBA, form variables are preceded with a v (i.e. vWorkerNm). The code below performs the same operations as DB Gateway's WKLIST and PRINTWK combined. Both the query and the results can be processed in the same file. When the form is loaded, IDBA sends the query to the database and then displays the results using the format indicated in the TYPE statement. The user then selects a name from the selection list and clicks the submit button. The selected name is passed to the page indicated in the form action, which prints the work list for the selected worker.

```
<FORM METHOD=POST ACTION="PrintWork.htm">
Please Select a Worker:
<!--Query: DSN=MUMHelpDesk:
SQL=Select WorkerNm from Workers:
TYPE=SELECT(vWorkerNm):
-->
<INPUT TYPE="SUBMIT"><INPUT TYPE="RESET">:
</FORM>
```

The PrintWork.htm file shown below will display the table of work orders for the selected name. The default template in DB Gateway handled this. When the page is loaded, vWorkerNm is replaced by the selected worker's name, the query is sent to the database, then the resulting data is displayed in table format.

```
The Work Orders for <STRONG>VAR(vWorkerNm)</STRONG> are <BR>
<!--Query: DSN=MUMHelpDesk:
```

```
SQL=SELECT DISTINCTROW Directory.LastName, NewWork.Problem,
NewWork.DueDate, NewWork.RefNum
FROM Directory RIGHT JOIN NewWork ON Directory.EmpID=NewWork.EmpID
WHERE NewWork.WorkerNm=VARQ(vWorkerNm)
AND NewWork.Status='Waiting': TYPE=TABLE:END>
```

Microsoft Internet Database Connector

The Microsoft Internet Database Connector (IDC) is a component of the Microsoft Internet Information Server (IIS). It requires Microsoft Windows NT Server 3.51 or higher. IDC uses two file types to interact with the database: .idc and .htx (HTML extension). The .idc file contains the information needed to execute the query against the database. The companion .htx file is used to format the data returned from the SQL query contained in the .idc file. The .idc file can be invoked from a hyperlink or form action. When the .idc file is requested, IDC formats the queries and sends them to the ODBC data source where the queries execute. IDC returns the result to the .htx file, which displays the result on the user's browser. Microsoft Access '97 and the IIS Add-In for Microsoft Access '95 automatically generate the .html, .idc and .htx files needed to query the database. The developer can quickly and easily generate the files to query the database, then customize them as needed.

IDC requires four files to select a worker and print the work list:

GetWorkers.idc is activated when the user selects the "Print Work Orders" menu option. The .idc file sends the query to the indicated data source. The data returned from the query is then passed to the indicated template, GetWorkers.htx.

```
DataSource: MUMHelpDesk
Template: GetWorkers.htx
SQLStatement:
+SELECT WorkerNm
+FROM Workers
```

The <%BeginDetail%> ... <%EndDetail%> section in the GetWorkers.htx file below contains the placeholders for the data returned from GetWorker.idc. The placeholders use the syntax <%filename%>. The code below displays the worker names in a selection list. The user selects a name, clicks the submit button, and the form action invokes GetWorkList.idc.

```
<FORM METHOD="POST" ACTION="GetWorkList.idc">
SelectWorker:
<SELECT NAME="WorkerNm">
<%BeginDetail%>
<OPTION><%WorkerNm%>
<%EndDetail%>
</SELECT>
</FORM>
```

GetWorkList.idc sends a query to the data source to select the work list for the selected worker. The query results are passed on to WorkRpt.htm indicated by the template field.

```
DataSource MUMHelpDesk
Template WorkRpt.htm
Required Parameters WorkerNm
SQLStatement
~SELECT NewWork.RefNum, NewWork.Date, Directory.LastName,
NewWork.Problem
~FROM Directory Right Join NewWork On Directory.EmpID = NewWork.EmpID
~WHERE NewWork.AssignedTo = "%WorkerNm%" AND NewWork.Status = "Waiting"
```

WorkRpt.htm displays the work list for the selected worker.

```
<TABLE>
<%BeginDetail%>
<TR>
<TD><%RefNum%></TD><TD><%Date%></TD><TD><%LastName%></TD>
<TD><%Problem%></TD>
<TR>
<%EndDetail%>
</TABLE>
```

Cold Fusion

Cold Fusion is a commercial product available from Allaire Corporation. A 30-day evaluation copy can be downloaded from <http://www.allaire.com>. A single user version is included in the Cold Fusion Web Database Construction Kit (see references). Cold Fusion can run under Windows NT IIS, Windows '95 using O'Reilly's WebSite, or any Web server that supports CGI.

Cold Fusion applications are built by combining HTML with Cold Fusion Markup Language (CFML). Cold Fusion tags begin with <CF ...>. Cold Fusion variables, functions and expressions are enclosed within #...#. CFML uses SQL to access ODBC data sources. All pages that use Cold Fusion have a .cfm extension. When the server receives a request for a .cfm page, it is sent to Cold Fusion, which formulates the queries and sends the queries to the data source. Cold Fusion uses the query results to translate the .cfm file into HTML. The resulting page is then forwarded to the requesting client.

The GetWorker.cfm code below displays links to work lists based on worker names. When the page is loaded, Cold Fusion sends the query to the data source. #WorkerNm# is replaced with the values returned from the query, displaying links for each of the worker names.

```
<CFQUERY NAME="GetWorkers" DATASOURCE="CCHelpDesk">
SELECT * FROM Workers
```

S

```

ORDER BY WorkerNm ASC
<CFQUERY>
<CFOUTPUT QUERY="GetWorkers">
<TR><TD>
<A HREF="PrintWorkList.cfm?WorkerName=#WorkerNm#">#WorkerNm#</A><BR>
</TD></TR>
</CFOUTPUT>

```

When PrintWorkList.cfm is invoked, it is passed the worker name (#WorkerNm#) as a URL parameter. When the page is loaded, the query uses the worker name to select the work list from the database. The data returned from the query is then printed in table format.

```

<CFQUERY NAME="GetWorkList" DATASOURCE="CCHelpDesk">
SELECT * FROM NewWork, Directory
WHERE Status='Waiting' AND Directory EmpID = NewWork EmpID
AND NewWork AssignedTo = #WorkerName#
ORDER BY Date ASC
</CFQUERY>
<TABLE>
<CFOUTPUT QUERY="GetWorkList">
<TR><TD>#LastName#</TD><TD>#Room#</TD><TD>#Bldg#</TD><TD>#Phone#</TD>
<TD>#DateFormat("#Date#", 'm/d/yy')#</TD><TD>#RefNum#</TD></TR>
<TR><TD COLSPAN=5>#Problem#<HR></TD></TR>
</CFOUTPUT>
</TABLE>

```

Conclusions

DB Gateway is too limited and too complex to be a useful tool. If you want a free product and do not have Windows NT server, IDBA looks like an acceptable alternative. If you already own Windows NT server, I recommend that you consider IDC. Even with the disadvantage of the file organization, IDC is worth further investigation. It is still fairly straightforward and has the added benefit of the Access IIS Add-In to automatically generate your .idc and .htx files. Cold Fusion is worth the investment if you need a Web programming development language. The features of the four tools are summarized in Figure 4.

Once I discovered Cold Fusion, I became addicted. I added many features to the Virtual Help Desk, and went on to find new applications. I implemented a numeric base conversion page for my Computer Architecture class. It uses built in Cold Fusion functions to randomly generate problems and solutions, then grade the students' answers. This gives the students much needed extra practice. I also created a multiple choice quiz generator. The user enters questions, answers and choices into an Access database, then the Quiz.cfm pages present the questions to the user, grade the quiz, and record the results in a database.

I am developing a course in World Wide Web programming for fall '97 which will provide me and my students with the opportunity to investigate additional tools.

Figure 4 Summary of the features of the four tools

Features	DB Gateway	IDBA	IDC	Cold Fusion
Free	X	X	Win NT	\$500
Technical Support				X
Reference Material			X	X
ODBC data source		X	X	X
All SQL supported		X	X	X
Queries and results in same page		X		X
If-Else construct			X	X
Any HTML output format	X		X	X
Convenient file location		X	X	X
Complete development environment				X
Wizard			X	X

REFERENCES

Forta, Ben (1997). The Cold Fusion Web Database Construction Kit. Que Corporation.

Swank, Mark and Drew Kittel (1996). World Wide Web Developers Guide. Sams Net.

Khurana Gunnit S. and Balbir S. Khurana (1996). Web Database Construction Kit. Waite Group Press.

IIS Add-In for Microsoft Access for Windows '95
<http://www.microsoft.com/Accessdev/ITK/AIReadMe.htm>

Job Forum. A Sample Internet Information Server Application that Stores and Retrieves Data from a Microsoft Access Database (1996)
<http://www.microsoft.com>

Microsoft Internet Information Server Overview for Reviewers (1996)
<http://www.microsoft.com>

PC Products for Accessing a Database Server via the World Wide Web
http://cscsun1.larc.nasa.gov/~beowulf/db/pc_products.html